Fundamentos de IA y Machine Learning

Lección 3: Modelos de Regresión y Clasificación I

eip ESCUELA INTERNACIONAL POSGRADOS

ÍNDICE

Modelos de Regresión y Clasificación I

Presentación y objetivos

Una vez que conocemos las técnicas de entrenamiento y evaluación, es el momento de exponer distintos modelos de aprendizaje automático y los algoritmos utilizados para su entrenamiento.

Objetivos

Al finalizar esta lección serás capaz de:

- Generar, entrenar e interpretar un modelo de regresión.
- Conocer la adaptación de la regresión para su uso en problemas de clasificación.
- Utilizar el algoritmo de los *k* vecinos más cercanos (*K-nearest neighbours, KNN*).
- Modelar e interpretar árboles de decisión.

1. Introducción

1.1. Contextualización de la lección

Como se observó en lecciones anteriores, las fases en las que se componen la resolución de un problema de aprendizaje automático son las siguientes.

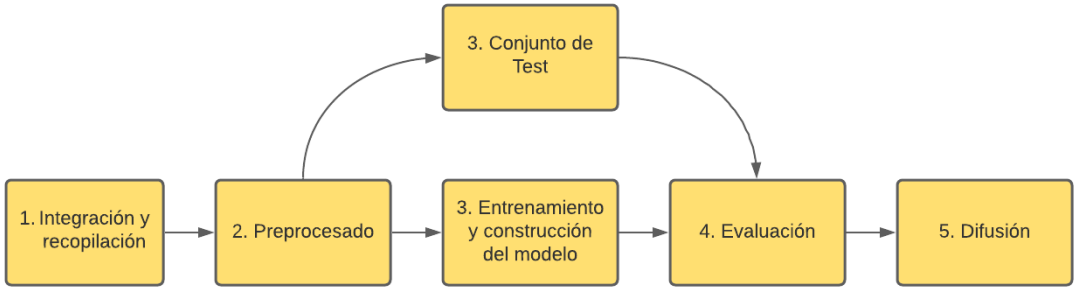

Figura 1.1: Fases de un problema de aprendizaje automático.

Una vez que se tiene el conocimiento necesario para recopilar los datos, preprocesarlos, elegir las técnicas de evaluación y las métricas de rendimiento, debemos determinar qué modelo es el más interesante.

Aunque existen muchos modelos en la literatura, en esta lección nos vamos a centrar en:

- **Regresión lineal simple y múltiple**.
- **Regresión logística**.
- **Algoritmo de los *k* vecinos más cercanos.**
- **Árboles de decisión.**

1.2. Formulación de un problema de regresión

Cualquier problema de regresión se puede formular de la siguiente forma:

- Disponemos de un espacio de entrada $\boldsymbol{X}$ compuesto por patrones etiquetados con $Y \subseteq \mathbb{R}$.
- Cada patrón se representa por un vector de características de dimensión K, $\boldsymbol{x} \in \boldsymbol{X} \subseteq \mathbb{R}^K$ y un valor continuo $y \in Y \subseteq \mathbb{R}$.
- El objetivo es aprender una función f que relacione los datos del espacio de entrada $\boldsymbol{X}$ al conjunto de valores continuos finito Y.
- El conjunto de entrenamiento $\boldsymbol{T}$ está compuesto de N patrones:

$$T = (\boldsymbol{x}_i, y_i) : \boldsymbol{x}_i \in \boldsymbol{X}, \boldsymbol{y_i} \in Y (i = 1, \dots, n), \text{ con } x_i = \left(x_{i,1}, x_{i,2}, \dots, x_{i,K}\right).$$

1.3. Formulación de un problema de clasificación

Análogamente, cualquier problema de clasificación se puede formular de la siguiente forma:

- Disponemos de un espacio de entrada $\boldsymbol{X}$ compuesto por patrones etiquetados con $C = \{C_1, C_2, \dots, C_Q\}$ donde Q es el número de clases.
- Cada patrón se representa por un vector de características de dimensión K, $\boldsymbol{x} \in \boldsymbol{X} \subseteq \mathbb{R}^K$ y una etiqueta de clase $y \in C$.
- El objetivo es aprender una función f que relacione los datos del espacio de entrada $\boldsymbol{X}$ al conjunto finito C.
- El conjunto de entrenamiento $\boldsymbol{T}$ está compuesto de N patrones:

$$T = (\boldsymbol{x}_i, y_i) : \boldsymbol{x}_i \in \boldsymbol{X}, \boldsymbol{y_i} \in C (i = 1, \dots, n), \text{ con } x_i = \left(x_{i,1}, x_{i,2}, \dots, x_{i,k}\right).$$

2. Regresión lineal

La regresión lineal es uno de los modelos más primitivos en comparación con otros modelos de aprendizaje automático más sofisticados que han ido surgiendo en los últimos años. Aun así, la regresión lineal sigue siendo muy utilizada en una gran variedad problemas en distintos campos de aplicación.

Además, sirve como introducción a los siguientes modelos del aprendizaje automático, ya que, como veremos más adelante, muchos de estos se consideran una generalización de la regresión lineal.

2.1. Fundamentos y modelo

La **regresión lineal simple** se basa en predecir un valor numérico Y, teniendo en cuenta un único valor de entrada X. Es decir, cada patrón tiene una única característica y se asume que existe una relación lineal entre X e Y. Matemáticamente, se puede escribir esta relación como:

$$Y \approx \beta_0 + \beta_1 X$$

siendo β_0 y β_1 el intercepto y la pendiente, respectivamente. Ambos parámetros son desconocidos y los tendremos que estimar haciendo uso del conjunto de entrenamiento.

En problemas reales y como vimos en la lección anterior, un patrón está compuesto por más de una característica de entrada. En este caso, estamos ante un modelo de **regresión lineal múltiple**. Para ello, debemos generalizar el modelo anterior teniendo en cuenta las K características de nuestro problema:

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_K X_K$$

La representación gráfica de un modelo regresión lineal múltiple con sus parámetros es la siguiente:

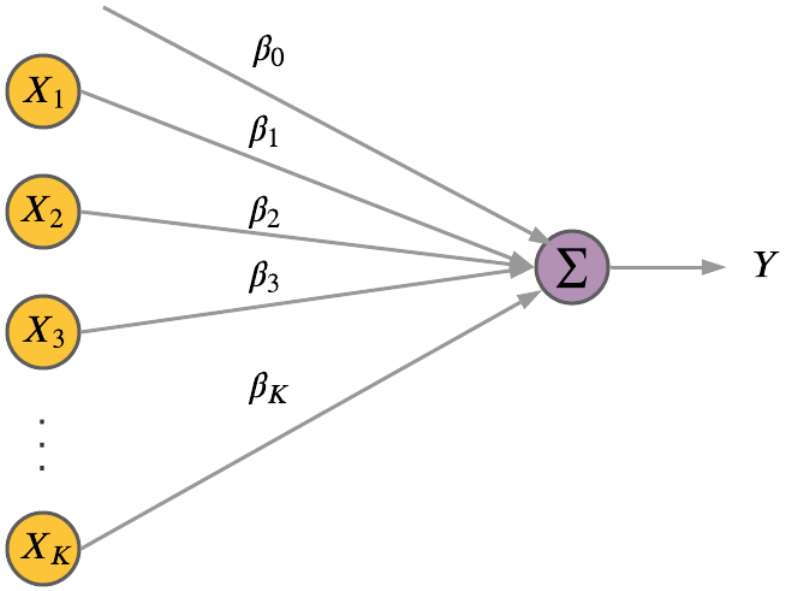

Figura 2.1: Modelo de regresión lineal múltiple.

2.2. Entrenamiento y estimación de los parámetros

Dado un conjunto de entrenamiento $(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \ldots, (\boldsymbol{x_n}, y_n)$, el objetivo es estimar los parámetros $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_K$, de forma que la línea resultante sea lo más "cercana" posible al conjunto de valores de Y.

Aunque existen diferentes formas de medir la cercanía entre un valor estimado y un valor real, el método más extendido es la minimización de los errores cuadráticos. Gráficamente, lo que se pretende es minimizar la distancia vertical entre el punto estimado por la recta y el punto real.

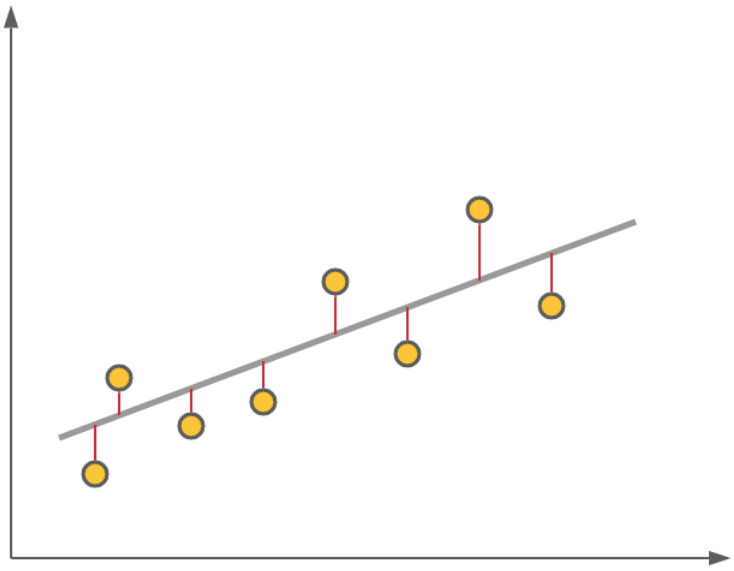

Figura 2.2: Minimización de los errores cuadráticos.

De esta forma, si tenemos $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_K x_{iK}$, como el valor estimado para el patrón $\boldsymbol{x_i}$. Se puede definir el residuo como $e_i = y_i - \hat{y}_i$. Por tanto, la suma del error cuadrático (*H*) será:

$$H = e_1^2 + e_2^2 + \cdots + e_n^2$$

$$H = \sum_{i=1}^{n} \left(y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_K x_{iK}\right)^2$$

Al tratarse de un problema de minimización, derivamos e igualamos a cero. Así, obtenemos que el cálculo específico de los coeficientes de regresión puede reducirse a utilizar la siguiente expresión:

$$\hat{\beta}_i = -\frac{A_{1,i+1}}{A_{1,1}}$$

Donde $A_{1,i+1}$ es el adjunto al elemento $a_{1,i+1}$ de la matriz de covarianzas:

$$\sum = \begin{pmatrix} S_y^2 & S_{y,x_1} & S_{y,x_2} & \dots \\ S_{x_1,y} & S_{x_1}^2 & S_{x_1,x_2} & \dots \\ S_{x_2,y} & S_{x_2,x_1} & S_{x_2}^2 & \dots \\ .. & \dots & \dots & \dots \end{pmatrix}$$

La ordenada en el origen $\hat{\beta}_0$ se calcula de manera inmediata:

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}_1 - \ \dots \ - \hat{\beta}_k \bar{x}_k$$

Simplificando las ecuaciones anteriores, en el caso de la regresión lineal simple, tendríamos que minimizar la siguiente expresión:

$$H \ = \ (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + \ (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \ \dots + \ (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$

De esta forma, se define:

$$\hat{\beta}_1 \ = \frac{S_{x,y}}{S_x^2}$$

y

$$\hat{\beta}_0 = \bar{y} \text{ - } \hat{\beta}_1 \bar{x}$$

Siendo $S_{x,y}$ la covarianza entre las variables x e y, S_x^2 es la varianza de la variable x, $\bar{y}$ es la media de los valores de y, y $\bar{x}$ es la media de los valores de $\bar{x}$.

Una vez que tenemos los coeficientes estimados, ya podemos generalizar y estimar valores sobre un conjunto no visto de *test* para medir el rendimiento de nuestro regresor.

2.3. Consideraciones importantes

Hay distintos aspectos que se deben tener en cuenta en el análisis de los modelos de regresión:

- Es recomendable que todas las variables de entrada estén escaladas en el mismo intervalo. De no ser así, tendríamos unos coeficientes difíciles de comparar entre sí.
- No existen hiperparámetros a determinar en el proceso de entrenamiento.
- Para determinar que atributo (variable de entrada) tiene mayor incidencia en la variable objetivo, se debe observar que $\hat{\beta}_k$ es el de mayor valor absoluto. Un valor positivo indicará que existe una fuerte relación lineal, y uno negativo una relación lineal inversa.
- Aquellos coeficientes cercanos a cero nos indican que la variable tiene poco peso en la predicción.
- Se debe medir el rendimiento del modelo en el conjunto de generalización, utilizando las métricas vistas en la lección 2 de la asignatura.

2.4. Caso de aplicación de regresión lineal simple y múltiple

El salario Y que gana un conjunto de personas, se ha basado en el número de horas X_1 y el número de incentivos X_2 que han sido capaces de conseguir. Hasta ahora, se ha aplicado de forma experimental y se tienen los siguientes datos de entrenamiento:

Y	113	118	127	132	136	144	138	146	156	149
X_1	20	20	25	25	30	30	30	40	40	40
X_2	1	2	1	2	1	2	3	1	2	3

Se pide hallar:

1. El modelo de regresión lineal simple de Y en función de X_1.
2. El modelo de regresión lineal simple de Y en función de X_2.
3. El modelo de regresión lineal múltiple de Y en función de X_1 y X_2.

4. Validar cada modelo con el siguiente conjunto de *test*.

Y	200	116	122	130	150	120	146	155	156	147
X_1	35	25	25	20	35	25	42	35	40	42
X_2	1	2	2	1	2	2	1	1	2	2

Solución

Antes de comenzar a calcular los modelos, podríamos escalar las variables de entrada en el rango [0, 1]. Para ello, $min(X_1) = 20$, $max(X_1) = 40$, $min(X_2) = 1$ y $max(X_2) = 3$. Aplicando la fórmula vista en la lección anterior se tiene:

Y	113	118	127	132	136	144	138	146	156	149
X_1	0	0	0.25	0.25	0.5	0.5	0.5	1	1	1
X_2	0	0.5	0	0.5	0	0.5	1	0	0.5	1

Además, es muy útil tener el calculado el vector de medias y la matriz de covarianzas:

$$\mu_{Y,X_1,X_2} = \begin{pmatrix} 135.9 \\ 0.5 \\ 0.4 \end{pmatrix}$$

$$\sum_{Y,X_1,X_2} = \begin{pmatrix} 168.69 & 4.525 & 1.84 \\ 4.525 & 0.1375 & 0.0375 \\ 1.84 & 0.0375 & 0.14 \end{pmatrix}$$

1. El modelo de regresión a estimar es $Y = \beta_0 + \beta_1 X_1$. Para ello, se calcula:

$$\hat{\beta}_1 = \frac{S_{X_1,Y}}{S_{X_1}^2} = \frac{4.525}{0.1375} = 32.91$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \overline{X_1} = 135.9 - 32.91 * 0.5 = 119.45$$

Así, el modelo estimado sería:

$$\boldsymbol{Y = 119.45 + 32.91X_1}$$

2. El modelo de regresión a estimar es $Y = \beta_0 + \beta_1 X_2$. Para ello, se calcula:

$$\hat{\beta}_1 = \frac{S_{X_2,Y}}{S^2_{X_2}} = \frac{1.84}{0.14} = 13.143$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \overline{X_2} = 135.9 - 13.143 * 0.4 = 130.643$$

Así, el modelo estimado sería:

$$\boldsymbol{Y = 130.643 + 13.143X_1}$$

3. El modelo de regresión a estimar es $Y = \beta_0 + \beta_1 X_1 + \beta_1 X_2$. Para ello, se calcula:

$$\hat{\beta}_i = -\frac{A_{1,i+1}}{A_{1,1}} \rightarrow \begin{cases} \hat{\beta}_1 = -\frac{A_{12}}{A_{11}} = -\frac{-0.5645}{0.018} = 31.636 \\ \hat{\beta}_2 = -\frac{A_{13}}{A_{11}} = -\frac{-0.0833}{0.018} = 4.628 \end{cases}$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \overline{X_1} - \hat{\beta}_2 \overline{X_2} = 135.9 - 31.636 * 0.5 - 4.628 * 0.4 = 118.231$$

$$\boldsymbol{Y = 118.231 + 31.636X_1 + 4.628X_2}$$

4. Antes de realizar las predicciones se deben escalar los datos teniendo en cuenta el máximo y el mínimo del conjunto de entrenamiento.

$\boldsymbol{Y}$	200	116	122	130	150	120	146	155	156	147
$\boldsymbol{X_1}$	0.75	0.25	0.25	0.5	0.75	0.25	1.1	0.75	1	1.1
$\boldsymbol{X_2}$	0	0.5	0.5	0	0.5	0.5	0	0	0.5	0.5

Las predicciones para cada modelo serán:

Y	200	116	122	130	150	120	146	155	156	147
$\hat{Y}_{M1}$	144.13	127.68	127.68	135.91	144.13	127.68	155.65	144.13	152.36	155.65
$\hat{Y}_{M2}$	130.64	137.21	137.21	130.64	137.21	137.21	130.64	130.64	137.21	137.21
$\hat{Y}_{M3}$	141.96	128.45	134.05	144.27	128.45	153.03	153.03	141.96	151.18	155.34

Para cada modelo calculamos los residuos $e_i = y_i - \hat{y}_i$:

e_{M1}	55.87	-11.68	-5.68	-5.91	5.87	-7.68	-9.65	10.87	3.64	-8.65
e_{M2}	69.36	-21.21	-15.21	-0.64	12.79	-17.21	15.36	24.36	18.79	9.79
e_{M3}	58.04	-12.45	-6.45	-4.05	5.73	-8.45	-7.03	13.04	3.82	-8.34

Y, por último, se calculan las métricas para cada modelo:

Métrica	*MAE*	*MSE*	*RMSE*	R^2
M1	12.548	371.735	19.280	0.3557
M2	20.471	722.990	26.889	0.2230
M3	12.741	399.004	19.975	0.3106

- Todas las métricas indican que el mejor modelo es el 1, mientras que el peor modelo es el 2.
- Observamos que la regresión lineal múltiple obtiene peores resultados que el modelo de regresión lineal simple 1, lo que indica que la variable X_2 introduce ruido al modelo y no es un buen predictor.
- Además, en el caso de la regresión lineal múltiple, $\hat{\beta}_1$ tiene un valor mucho mayor que $\hat{\beta}_2$, lo que indica que la variable X_1 tiene mayor importancia en la predicción que X_2, corroborando lo comentado en el punto anterior.
- Aun así, se puede observar que los modelos no son muy buenos ya que se cometen errores relativamente altos y el coeficiente de determinación es cercano a 0.

3. Regresión logística

Cuando la variable de salida no es numérica sino categórica estamos ante un problema de clasificación. Aunque una regresión lineal no puede ser utilizada para tal fin, muchos de los métodos de clasificación se basan en asignar probabilidades de pertenencia de cada patrón a cada clase. Es por ello, que funcionan bajo el mismo fundamento.

3.1. Fundamentos y modelo

Dos de las principales razones por las que no se puede usar la regresión lineal en problemas de clasificación son:

1. No puede dar una respuesta cualitativa, ya que si intentamos asignar a cada clase un valor, estamos asumiendo una distancia que no sabemos.
2. Las probabilidades $P(Y|X)$, probabilidad de que un patrón pertenezca a una clase, no serán significativas. Por ejemplo, podríamos tener valores negativos o superiores a uno, sin embargo, la probabilidad es un valor comprendido en el rango [0, 1].

Una modificación de la regresión lineal es la regresión logística que modela la probabilidad $P(X)$, usando una función que da una salida acotada entre 0 y 1 para todos los valores de X.

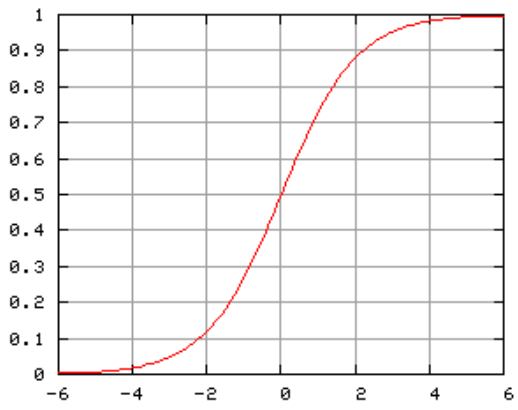

Figura 3.1: Función logística.

En un problema de clasificación binaria con una única variable predictora (**regresión logística simple**), la probabilidad $p(X)$ de pertenencia de un patrón a la primera clase de entre dos se formula de la siguiente forma:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

La probabilidad de pertenencia a la otra clase será su complementaria:

$$1 - p(X) = \frac{1}{1 + e^{\beta_0+\beta_1 X}}$$

En el caso de tener más de una característica predictora, estamos ante la **regresión logística múltiple** y la probabilidad anterior se define como:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\ldots+\beta_K X_K}}{1 + e^{\beta_0+\beta_1 X_1+\ldots+\beta_K X_K}}$$

Cuando el problema al que nos enfrentamos es multiclase, es decir, la variable a predecir tiene Q categorías. La formulación queda como sigue:

$$p(Y = q|X) = \frac{e^{\beta_{q0}+\beta_{q1} X_1+\ldots+\beta_{qK} X_K}}{1 + \sum_{l=1}^{Q-1} e^{\beta_{l0}+\beta_{l1} X_1+\ldots+\beta_{lK} X_K}}$$

Si observamos, aquí generaremos tantos modelos como clases tengamos menos una ($Q - 1$ modelos). De forma que, para cada clase tenemos un modelo que nos indica la probabilidad de pertenencia de cada patrón a dicha clase, y la probabilidad de pertenencia de la última clase se calcularía como 1 menos la suma del resto.

La representación gráfica de un modelo de regresión logística múltiple multiclase es la siguiente:

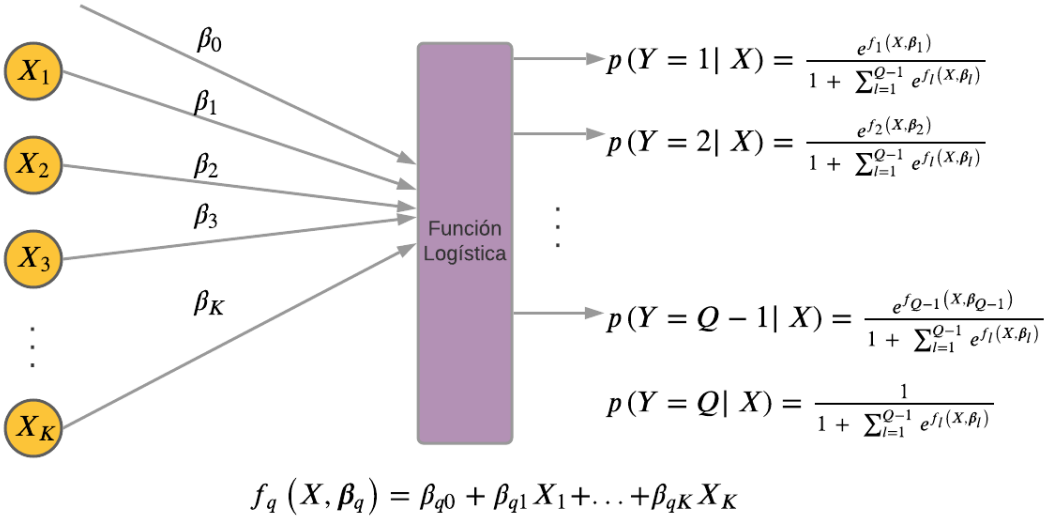

Figura 3.2: Modelo de regresión logística múltiple multiclase.

Podemos expresar la regresión logística de forma lineal. Manipulando la ecuación anterior (utilizaremos un problema binario por simplicidad) se tiene:

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1X_1+\cdots+\beta_KX_K}$$

Y aplicando logaritmos:

$$\ln\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1X_1 + \cdots + \beta_KX_K$$

El término de la izquierda se denomina *logit*.

3.2. Entrenamiento y estimación de los parámetros

De nuevo, los coeficientes β_0 y β_1 son desconocidos y deben ser estimados haciendo uso del conjunto de entrenamiento. El método más utilizado es el denominado método de máxima verosimilitud (*maximum likelihood*) ya que tiene mejores propiedades estadísticas asociadas.

Lo que trata de buscar este método es β_0 y β_1 de forma que $p(X)$ sea lo más cercano a uno cuando el patrón pertenece a la clase positiva y cercano a cero cuando no lo es. Formalmente:

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y'_i=0} \left(1 - p(x'_i)\right)$$

Muchos de los métodos de aprendizaje automático, utilizan el método de máxima verosimilitud. De hecho, el método de mínimos cuadrados utilizado en la regresión lineal es un caso especial de este método.

Como siempre, una vez que se han estimado los parámetros del modelo, se debe generalizar haciendo uso de un conjunto de *test*, para obtener el rendimiento de nuestro clasificador.

3.3. Consideraciones importantes

Los aspectos más importantes de la regresión logística son:

- La función logística nos proporciona un valor de pertenencia de cada patrón a cada clase.
- Esta función hace que la suma de las probabilidades de pertenencia a cada clase sea uno.
- El patrón tendrá como clase predicha aquella clase cuya probabilidad de pertenencia sea máxima.
- Al igual que en la regresión lineal, es recomendable que todas las variables de entrada estén escaladas en el mismo intervalo. De no ser así, tendríamos unos coeficientes difíciles de comparar entre sí.
- No existen hiperparámetros a determinar en el proceso de entrenamiento.

- Para determinar que atributo (variable de entrada) tiene mayor incidencia en la variable objetivo, se debe observar que $\hat{\beta}_k$ es el de mayor valor absoluto.
- Aquellos coeficientes cercanos a cero nos indican que la variable tiene poco peso en la predicción.
- Se debe medir el rendimiento del modelo en el conjunto de generalización, utilizando las métricas vistas en la lección 2 de la asignatura.

3.4. Ejemplo de aplicación de un modelo de regresión logística múltiple

Haciendo uso de la base de datos *Iris* vista anteriormente, se ha ejecutado un algoritmo para estimar los parámetros de una regresión logística múltiple. Como el problema tiene tres clases: setosa (1), versicolor (2) y virgínica (3), se han generado los dos modelos siguientes:

$$f_1(\boldsymbol{X}, \beta_1) = 8.1743 + 21.8065X_1 + 4.5648X_2 - 26.3083X_3 - 43.887X_4$$

$$f_2(\boldsymbol{X}, \boldsymbol{\beta_2}) = 42.637 + 2.4652X_1 + 6.6809X_2 - 9.4293X_3 - 18.2859X_4$$

Se pide:

1. Hallar las predicciones del modelo para el siguiente conjunto de *test*.
2. Evaluar el rendimiento del clasificador en dicho conjunto.

Patrón	X_1	X_2	X_3	X_4	Clase
1	4.6	3.2	1.4	0.2	1
2	5.3	3.7	1.5	0.2	1
3	5.7	4.4	1.5	0.4	1
4	5.0	3.5	1.6	0.6	2
5	5.5	2.5	4.0	1.3	2
6	5.7	3.0	4.2	1.2	2
7	5.7	2.8	4.1	1.3	3
8	5.8	2.7	5.1	1.9	3
9	6.3	2.5	5.0	1.9	3
10	5.9	3.0	5.1	1.8	3

Solución:

1. Se evaluan la funciones $f_1(\boldsymbol{X}, \beta_1)$ *y* $f_2(\boldsymbol{X}, \beta_2)$, se aplica la transformación logística para las probabilidades de las tres clases, y la clase predicha será el máximo de las tres.

P	e^{f_1}	e^{f_2}	$\sum e^{f_l}$	$p(Y=1\|X)$	$p(Y=2\|X)$	$p(Y=3\|X)$	Clase predicha	Clase Real
1	4.47E+33	2.54205E+25	4.4693E+33	0.99999999	5.68781E-09	0	1	1
2	1.34E+40	1.56982E+27	1.3435E+40	1	1.16848E-13	0	1	1
3	3.11E+41	1.16638E+28	3.1057E+41	1	3.75565E-14	0	1	1
4	1.33E+28	5.10826E+22	1.3307E+28	0.99999616	3.83863E-06	0	1	2
5	1.3E-10	90127.44471	90127.4447	1.4413E-15	0.999988905	1.1095E-05	2	2
6	5.17E-10	631978.087	631978.087	8.1869E-16	0.999998418	1.5823E-06	2	2
7	2.88E-09	426496.739	426496.739	6.7603E-15	0.999997655	2.3447E-06	2	3
8	2.22E-31	0.000386241	0.00038624	2.2235E-31	0.000386092	0.99961391	3	3
9	6.74E-26	0.000894094	0.00089409	6.7313E-26	0.000893295	0.99910671	3	3
10	6.24E-28	0.022830224	0.02283022	6.0978E-28	0.022320639	0.97767936	3	3

2. Una vez se han realizado las predicciones, se calcula la matriz de confusión, y las métricas CCR y $Kappa$.

$$\begin{pmatrix} 3 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 3 \end{pmatrix}$$

$$CCR = \frac{1}{n}\sum_{j=1}^{J} n_{jj} = 80\%$$

$$Kappa = \frac{p_0 - p_e}{1 - p_e} = \frac{CCR - \frac{1}{n^2}\sum_{j=1}^{J} n_{j\cdot}\, n_{\cdot j}}{1 - \frac{1}{n^2}\sum_{j=1}^{J} n_{j\cdot}\, n_{\cdot j}} = \frac{0.8 - 0.33}{1 - 0.33} = 0.70$$

Para cada una de las clases consideramos como positiva dicha clase y negativa el resto. Así, se forman las siguientes matrices de confusión:

Clase 1: $\begin{pmatrix} 3 & 0 \\ 1 & 6 \end{pmatrix}$ Clase 2: $\begin{pmatrix} 2 & 1 \\ 1 & 6 \end{pmatrix}$ Clase 3: $\begin{pmatrix} 3 & 1 \\ 0 & 6 \end{pmatrix}$

A partir de aquí, calculamos las métricas para cada una de las clases:

Clase	$\boldsymbol{Sensibilidad}$	$\boldsymbol{FP\ Rate}$	$\boldsymbol{Especifidad}$	$\boldsymbol{Precision}$	$\boldsymbol{F-Score}$
1	1	0.143	0.857	0.750	0.857
2	0.667	0.143	0.857	0.667	0.667
3	0.750	0	1	1	0.857
Promedio	0.806	0.095	0.905	0.806	0.794

4. Algoritmo de los k-vecinos más cercanos

El algoritmo de los k-vecinos más cercanos (*k-Nearest Neighbours, KNN*) es un algoritmo de clasificación y de regresión que opera bajo un principio muy simple: etiquetar a los patrones del conjunto de generalización dependiendo de los patrones más cercanos a ellos.

4.1. Fundamentos, modelo y entrenamiento

Este algoritmo se basa en clasificar o bien predecir un valor numérico mirando los k vecinos más cercanos a él. Por ejemplo, en un problema para clasificar gorilas y perros, se toman dos características de entrada: altura y peso.

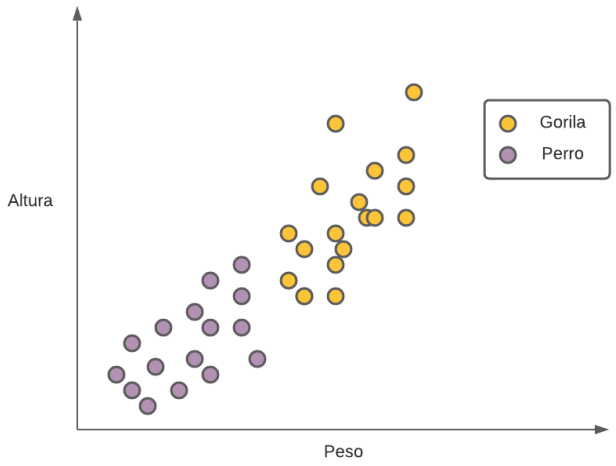

Figura 4.1: Clasificación de gorilas y perros.

El algoritmo no tiene entrenamiento de ningún modelo, sino que la predicción se basa en mirar un dato que no pertenece al conjunto de entrenamiento, observar los vecinos más cercanos, y en función de sus etiquetas, dar una respuesta.

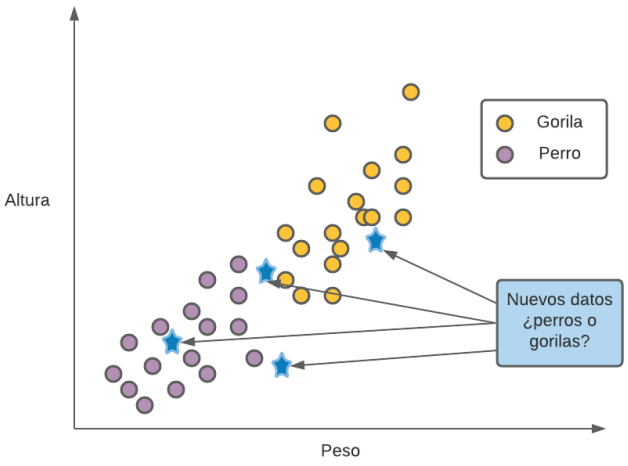

Figura 4.2: Predicción en un conjunto no visto.

Concretamente, para cada patrón X el algoritmo calcula la distancia con respecto a todos los puntos del conjunto de entrenamiento. Posteriormente, ordena los puntos en orden creciente de la distancia con respecto a X. Finalmente, predice la clase mayoritaria de los k vecinos más cercanos si se trata de una clasificación, o bien, calcula la media de sus valores para un problema de regresión.

Se puede observar que en este caso tenemos dos parámetros muy importantes en el algoritmo. Por un lado, el tipo de distancia que se va a usar influye en la predicción. Hay distintos tipos de distancias en la literatura, pero las distancias más conocidas entre dos puntos son:

- **Distancia euclídea:** raíz cuadrada de las diferencias al cuadrado de sus coordenadas.

$$d_E(X_1, X_2) = \sqrt{\sum_{i=1}^{K} (X_{1i} - X_{2i})^2}$$

Distancia Manhattan: suma de las diferencias absolutas de sus coordenadas.

$$d_M(X_1, X_2) = \sum_{i=1}^{K} |X_{1i} - X_{2i}|$$

Distancia Ajedrez: mayor de sus diferencias a lo largo de cualquiera de sus dimensiones coordenadas.

$$d_A(X_1, X_2) = max_{i=1}^{K}(|X_{1i} - X_{2i}|)$$

Por otro lado, es importante establecer el parámetro k, ya que su valor va a provocar que un nuevo patrón sea clasificado en una clase u otra. A valores pequeños, sólo se tendrán en cuenta patrones muy cercanos, pero se ignorarán algunos que podrían ser importantes, mientras que a valores muy grandes se tendría mucho ruido en la clasificación o regresión. Lo ideal es encontrar el parámetro ideal que refleje un equilibrio entre ambos extremos.

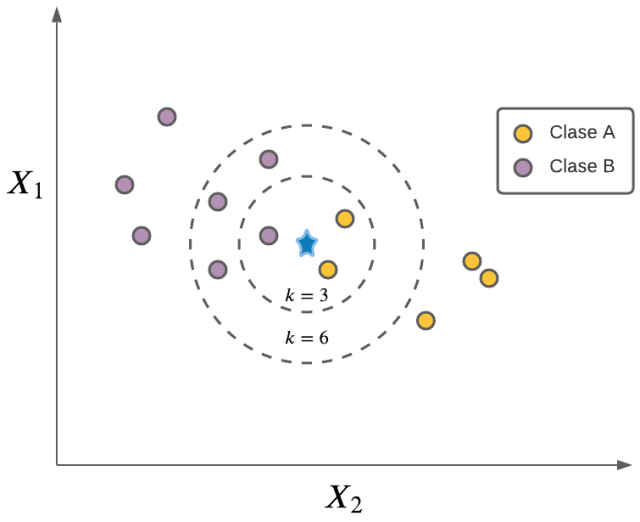

Figura 4.3: Cambio del parámetro k.

En el ejemplo de la Figura 4.3. se observa que si el parámetro $k = 3$, la clase predicha sería la Clase A, mientras que si establecemos $k = 6$, la clase predicha sería la Clase B.

4.2. Consideraciones importantes

Entre las características y los aspectos más importantes a tener en cuenta, destacamos:

- Es un algoritmo cuyo principio de funcionamiento es simple.
- No tiene entrenamiento.
- Funciona tanto para regresión como para clasificación.
- Se necesitan estimar el parámetro k y la medida de distancia.
- Es un algoritmo con alto coste computacional para la predicción, por lo que no es recomendable en base de datos muy grandes.
- No es bueno cuando el número de características de los patrones es muy alto.
- Con características categóricas el rendimiento es peor.

4.3. Ejemplo de aplicación del algoritmo *KNN*

Para un problema de clasificación binaria y con tres características se han recogido los siguientes patrones de entrenamiento.

Patrón	X_1	X_2	X_3	Clase
1	4.6	3.2	1.4	1
2	5.3	3.7	1.5	1
3	5.7	4.4	1.5	1
4	5.0	3.5	1.6	2
5	5.5	2.5	4.0	1
6	5.7	3.0	4.2	2
7	5.7	2.8	4.1	2
8	5.8	2.7	5.1	1
9	6.3	2.5	5.0	2
10	5.9	3.0	5.1	2

Se pide:

1. Hallar las predicciones del modelo para el siguiente conjunto de *test*, teniendo en cuenta el valor de $k = 3$ vecinos y la distancia euclídea.
2. Evaluar el rendimiento del clasificador en dicho conjunto.

Patrón	X_1	X_2	X_3	Clase
1	5	3.5	1.7	1
2	4.3	2.8	1.5	1
3	2.7	4.5	1.2	1
4	5.0	4.2	1.3	1
5	6.3	2.5	4.1	1
6	5.2	3.0	4.5	2
7	4.5	3	4.2	2
8	5.9	2.9	5.2	2
9	5	2.4	5.1	2
10	4.5	3.2	5.0	2

Solución:

1. Se calcula la distancia de cada patrón de *test* a los de entrenamiento, se seleccionan los tres más cercanos y se determina la clase predicha.

	Patrón entrenamiento											
P. test	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Clase predicha**	**Clase Real**
1	**0,5831**	**0,4123**	1,1576	**0,1**	2,5573	2,6439	2,5962	3,5833	3,6851	3,5525	1	1
2	**0,5099**	**1,3454**	2,126	**0,995**	2,7893	3,048	2,953	3,9013	4,0423	3,9446	1	1
3	**2,3108**	**2,7368**	3,0166	**2,5397**	4,4362	4,5	4,5056	5,2972	5,6036	5,2631	1	1
4	1,0817	**0,6164**	**0,755**	**0,7616**	3,2296	3,2156	3,2078	4,1629	4,2743	4,0853	1	1
5	3,2665	3,0332	3,2757	2,99	**0,8062**	**0,7874**	**0,6708**	1,1358	0,9	1,1874	2	1
6	3,1639	3,0822	3,3481	2,9496	**0,7681**	**0,5831**	**0,6708**	0,9	1,3077	0,922	2	2
7	2,8089	2,9017	3,2696	2,6944	**1,1358**	**1,2**	**1,2207**	1,6093	2,0322	1,6643	2	2
8	4,0274	3,8328	3,9975	3,759	1,3266	1,0247	1,1225	**0,2449**	**0,6**	**0,1414**	2	2
9	3,8066	3,8393	4,1773	3,6688	**1,2124**	1,2884	1,2845	**0,8544**	1,3077	**1,0817**	1	2
10	3,6014	3,6249	3,8897	3,4496	1,578	**1,456**	1,5524	**1,3964**	1,9313	**1,4177**	2	2

2. Se construye la matriz de confusión y se calculan las métricas:

$$\begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}$$

$$CCR = \frac{TP + TN}{N} = 0.8$$

$$Sensibilidad = \frac{TP}{TP + FN} = 0.8$$

$$FP\ Rate = \frac{FP}{TN + FP} = 0.2$$

$$Especifidad = \frac{TN}{TN + FP} = 0.8$$

$$Precisión = \frac{TP}{TP + FP} = 0.8$$

$$F1 - Score = \frac{2TP}{2TP + FP + FN} = 0.8$$

5. ÁRBOLES DE DECISIÓN

Los árboles de decisión son modelos muy intuitivos utilizados tanto en regresión como en clasificación. Debido a su interpretabilidad, son muy utilizados en distintas aplicaciones del mundo real. Existen distintos tipos de árboles de decisión, sin embargo, todos funcionan bajo fundamentos similares.

5.1. Fundamentos y modelo

Se denominan árboles de decisión ya que la representación gráfica de las funciones aprendidas son árboles dirigidos con un nodo raíz y uno o varios nodos hoja. Cada **nodo** no terminal especifica una toma de decisión de algún atributo de la instancia o patrón, cada **rama** corresponde a un posible valor del atributo, y cada **nodo terminal** u **hoja** indica la clase en la que se clasifica o el valor que se predice.

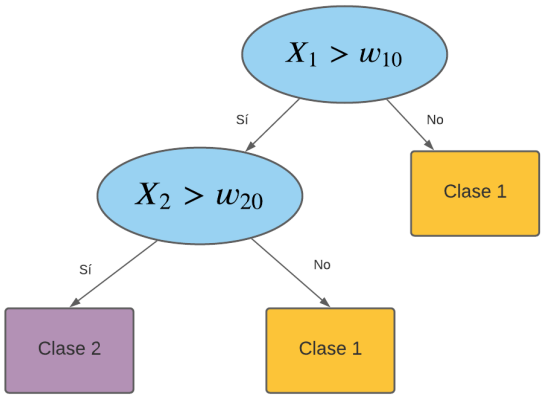

Figura 5.1: Representación de un árbol de decisión.

El árbol representa una disyunción de conjunciones de restricciones sobre los valores de los atributos de las instancias. De este modo, un camino es una conjunción de un elemento en *test*, mientras que todo el árbol sería una disyunción de estas conjunciones. En la Figura 5.1 el árbol para la clase 1 es: $(X_1 > w_{10}) \vee (X_1 < w_{10} \wedge X_2 > w_{20})$.

En definitiva, un árbol es un conjunto de reglas:

- **R1: Si** $X_1 > w_{10}$ **Entonces** Clase 1.
- **R2: Si** $X_1 < w_{10}$ **Y** $X_2 > w_{20}$ **Entonces** Clase 1.
- **R3: Si** $X_1 < w_{10}$ **Y** $X_2 < w_{20}$ **Entonces** Clase 2.

Esto se interpreta en una división del espacio en regiones etiquetadas con una clase y estas regiones son hiperrectángulos. Por simplicidad, a continuación, se muestra la división del espacio en rectángulos de dos dimensiones resultante del árbol anterior.

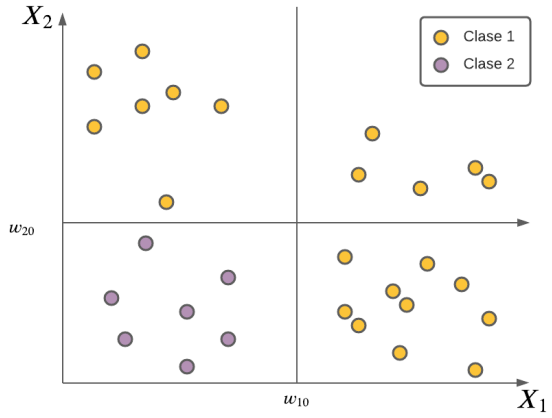

Figura 5.2: Regiones divididas en rectángulos.

5.2. Entrenamiento y estimación de los parámetros

El entrenamiento consiste en utilizar el conjunto de *train* para generar el árbol de clasificación o regresión. La idea es construir un árbol de arriba abajo empezando por el nodo raíz. Aunque existen muchas variantes de árboles de decisión, el proceso se resume en los siguientes pasos:

1. Se selecciona el mejor atributo como nodo.
2. Se abre el árbol para cada posible valor del atributo.
3. Los ejemplos se van clasificando en los nodos apropiados.

4. Repetir el proceso usando los ejemplos asociados con el nodo en el que estemos.
5. Parar cuando se satisface la condición de parada, por ejemplo, el árbol clasifica correctamente los ejemplos o se han usado todos los atributos.
6. Etiquetar el nodo hoja con la clase de los ejemplos.

En todo este proceso, y lo que en gran medida diferencia un tipo de árbol de decisión de otro es el cómo se selecciona el mejor atributo en cada nodo del árbol. Algunas de las técnicas más utilizadas son:

Ganancia de información mutua (ID3): para cada atributo se busca maximizar la expresión $I(C, X_i) = H(C) - H(C|X_i)$.

$$H(C) = -\sum_{c=1}^{Q} p(c) \log_2 p(c)$$

$$H(C|X) = -\sum_{c=1}^{Q} p(x|c)\, log_2\, p(c|x) = -\sum_{c=1}^{Q}\sum_{x} p(c,x)\, log_2\, p(c|x)$$

Maximizar la ratio de ganancia (C4.5): para cada atributo se maximiza la expresión

$$I(C, X_i)/H(X_i).$$

Índice Gini (CART): nos mide la probabilidad de no sacar dos registros de la misma clase del nodo. Se debe minimizar la expresión

$$GINI(X_i) = 1 - \sum_{c=1}^{Q} p(c)^2$$

Durante el entrenamiento de árboles de decisión, uno de los principales problemas es el sobreentrenamiento. Si hacemos un árbol que crece hasta que clasifique correctamente todos los ejemplos de entrenamiento, hace que el modelo no sea capaz de generalizar.

Para ello, se aplican dos técnicas de poda del árbol, es decir, técnicas que permiten eliminar nodos, de forma que los ejemplos de esos nodos se reajustan en nodos superiores produciendo un árbol que generaliza mejor:

- **Pre-poda:** para de aumentar el árbol antes de que alcance el punto en el que clasifica perfectamente los ejemplos de entrenamiento. Es difícil estimar cuándo hacerlo. Se aplican métodos como tests estadísticos para estimar si expandiendo un nodo particular es probable producir una mejora más allá del conjunto de entrenamiento.
- **Post-poda:** permitir que se sobreajuste los datos, y después podar el árbol reemplazando subárboles por una hoja. Es mejor en la práctica, pero también es más costoso computacionalmente.

De aquí se determina la necesidad de optimizar al menos tres hiperparámetros, aunque sí es cierto que los árboles decisión no son muy sensibles a hiperparámetros. Solamente uno de ellos es el más importante, que sería decidir si se aplica poda o no. Si se decide realizar poda, otro hiperparámetro sería el factor de poda que nos indica la fuerza con la que se aplicará la poda. Y, por último, tenemos el mínimo número de ejemplos necesarios para crear un nodo hoja.

5.3. Consideraciones importantes

A modo de resumen y como aspectos importantes podemos destacar:

- Los árboles de decisión, como su nombre indica, se representan mediante árboles compuestos por nodos no terminales, ramas y nodos hoja.
- Dada su interpretabilidad, son aplicados en muchos problemas del mundo real.
- Son robustos a datos con ruido.
- Algunos árboles de decisión permiten valores perdidos.
- Permiten valores categóricos, por lo que no hay necesidad de binarizar las variables de entrada.

- Son modelos no lineales, que dividen el espacio en hiperrectángulos.
- No son muy sensibles a hiperparámetros.
- Pueden sobreentrenar, por lo que normalmente es necesario realizar una poda.
- Se debe evitar que crezcan demasiado para no perder la interpretabilidad de ellos.

Importante

Es importante mirar y aprender bien lo que son los *RandomForests*. Para ello, consultar en enlace: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

5.4. Ejemplo de aplicación de árboles de decisión

Se ha aplicado un algoritmo para entrenar el siguiente modelo de árbol de decisión, para determinar si se puede jugar (Sí) o no al tenis (No):

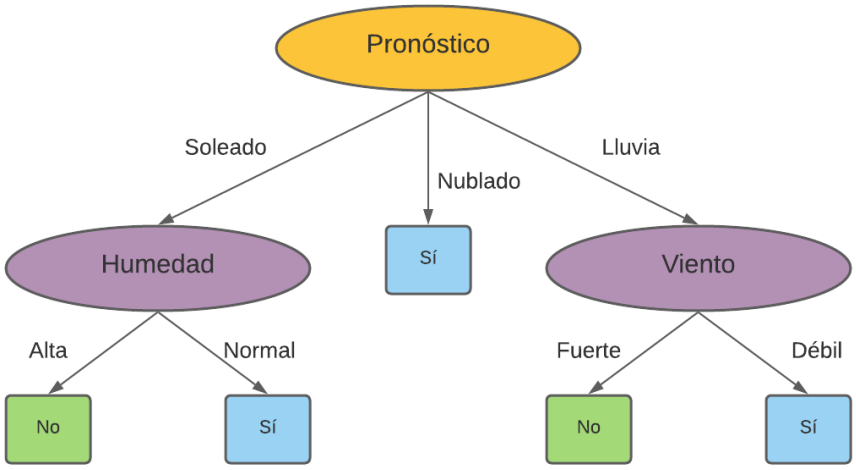

Figura 5.3: Árbol de decisión para jugar o no al tenis.

Se pide:

1. Analizar el árbol de decisión.
2. Hallar las predicciones del modelo para el conjunto de datos siguiente.
3. Evaluar el rendimiento del clasificador en dicho conjunto.

Día	Pronóstico	Temperatura	Humedad	Viento	Jugar
1	Soleado	Calor	Alta	Débil	No
2	Soleado	Calor	Alta	Fuerte	No
3	Lluvia	Media	Alta	Débil	Sí
4	Lluvia	Frío	Normal	Débil	Sí
5	Nublado	Frío	Normal	Fuerte	Sí
6	Soleado	Media	Alta	Débil	No
7	Lluvia	Media	Normal	Débil	No
8	Soleado	Media	Normal	Fuerte	Sí
9	Nublado	Calor	Normal	Débil	Sí
10	Lluvia	Media	Alta	Débil	No

Solución:

1. El árbol de decisión tiene como nodo raíz, y por tanto característica más importante en el modelo, la variable Pronóstico. Se puede observar que se ha descartado la característica Temperatura, por lo que podemos concluir que dicha característica no nos aporta información en nuestra predicción. El árbol resultante consta de 5 nodos hoja, tres nodos no terminales incluyendo la raíz, y 7 ramas que corresponden a los posibles valores de los tres atributos utilizados por el modelo.

2. Realizamos las predicciones utilizando el árbol generado:

Día	Pronóstico	Temperatura	Humedad	Viento	Predicción	Jugar
1	Soleado	Calor	Alta	Débil	No	No
2	Soleado	Calor	Alta	Fuerte	No	No
3	Lluvia	Media	Alta	Débil	Sí	Sí
4	Lluvia	Frío	Normal	Débil	Sí	Sí
5	Nublado	Frío	Normal	Fuerte	Sí	Sí
6	Soleado	Media	Alta	Débil	No	No
7	Lluvia	Media	Normal	Débil	Sí	No
8	Soleado	Media	Normal	Fuerte	Sí	Sí
9	Nublado	Calor	Normal	Débil	Sí	Sí
10	Lluvia	Media	Alta	Débil	Sí	No

3. Se construye la matriz de confusión y se calculan las métricas:

$$\begin{pmatrix} 5 & 0 \\ 2 & 3 \end{pmatrix}$$

$$CCR = \frac{TP + TN}{N} = 0.8$$

$$Sensibilidad = \frac{TP}{TP + FN} = 1$$

$$FP\ Rate = \frac{FP}{TN + FP} = 0.4$$

$$Especifidad = \frac{TN}{TN + FP} = 0.6$$

$$Precisión = \frac{TP}{TP + FP} = 0.71$$

$$F1 - Score = \frac{2TP}{2TP + FP + FN} = 0.83$$

6. PUNTOS CLAVE

Una vez que hemos desarrollado los puntos más importantes de la primera sesión de los modelos de regresión y clasificación, estaremos capacitados para:

- Ajustar los parámetros de un modelo de regresión lineal, realizar predicciones en el conjunto de generalización e interpretar sus resultados.
- Comprender y saber aplicar la regresión logística para problemas de clasificación.
- Aplicar el algoritmo *KNN* tanto para clasificación como para regresión.
- Generar e interpretar modelos de árboles de decisión.
- Utilizar las métricas de rendimiento para comparar los distintos modelos.